#### 序列化

**概述**：Java提供一种对象序列化的机制，用一个字节序列可以表示一个对象，该字节序列包含该对象的数据、对象的类型、对象中存储的属性等信息。字节序列写出到文件之后，相当于文件中持久保存了一个对象的信息；反之该字节序列还可以从文件中读取回来，重构对象，对它进行反序列化。对象的数据、对象的类型和对象中存储的数据信息，都可以用来在内存中创建对象。

##### 反序列化操作1

**序列化**：把对象以流的方式写入到文件中保存，叫写对象，也叫对象的序列化

ObjectOutputStream

特有成员方法：void writeObject(Object obj)：将指定的对象写入objectOutputStream中

**反序列化**：把文件中保存的对象，以流的方式读取出来，也叫对象的反序列化

反序列化前提：

类必须实现Serializable接口，Serializable接口给需要序列化的类，提供了一个序列化版本号，serializableID该版本号的目的在于验证序列化对象和对应类是否版本匹配

对于jvm可以反序列化对象，必须有类对应的class文件，否则会抛出ClassNotFoundException异常；

扩展：被static修饰的成员变量不能被序列化；

##### 反序列化操作2

当jvm反序列化对象时，能找到class文件，但是class文件在序列化对象后发生了修改，那么反序列化操作也会失败，抛出一个InvalidClassException异常；

**产生原因**：

该类的序列化版本号与流中读取的类描述符的版本号不匹配

该类包含未知数据类型

该类没有可访问的无参构造方法；

解决方式，给实现类显示定义一个序列化版本号；

static final long *serialVersionUID；*

##### Transient关键字

如果实现反序列化接口的类中有一个属性不需要序列化的，则该属性必须注明是瞬态的，使用transient关键字修饰；